

CLAIMS

What is claimed is:

1. A multi-thread accumulation circuit that supports a plurality of threads, comprising:

5 a first operation unit operably coupled to receive a first operand and a second operand corresponding to an operation code issued by a selected thread of the plurality of threads, wherein the operation unit combines the first and second operands to produce a first operation result corresponding to the selected thread;

10 a plurality of accumulation registers operably coupled to the first operation unit, wherein each accumulation register of the plurality of accumulation registers corresponds to one of the plurality of threads, wherein a selected accumulation register that corresponds to the selected thread stores the first operation result corresponding to the
15 selected thread; and

20 a selection block operably coupled to the plurality of accumulation registers and the first operation unit, wherein the selection block selects the second operand provided to the first operation unit from a set of potential operands, wherein the set of potential operands includes contents of each accumulation register of the plurality of accumulation registers.

2. The multi-thread accumulation circuit of claim 1 further comprises a control
25 block operably coupled to the selection block and the plurality of accumulation registers, wherein the control block receives information based on the operation code and generates control information provided to the plurality of accumulation registers and the selection block, wherein the control information provided to the plurality of accumulation registers causes the selected accumulation register to store the result corresponding to the selected thread when the operation code corresponds to an accumulate operation.

30

3. The multi-thread accumulation circuit of claim 2, wherein when the operation code corresponds to an accumulate operation, the control block provides the control information to the selection block such that the selection block selects a current value stored in the selected accumulation register as the second operand.

5

4. The multi-thread accumulation circuit of claim 3, wherein the first operation unit performs an addition operation such that the result of an accumulate operation is a sum of the current value stored in the selected accumulation register and the first operand.

10 5. The multi-thread accumulation circuit of claim 4 further comprises a second operation unit operably coupled to the first operation unit, wherein the second operation unit is operably coupled to receive a third operand and a fourth operand, wherein the second operation unit combines the third and fourth operands to produce a second operation result, wherein the second operation result is provided to the first operation unit
15 as the first operand.

6. The multi-thread accumulation circuit of claim 5, wherein the second operation unit performs multiplication operations such that a plurality of multiply and accumulate functions are supported for the plurality of threads by the multi-thread accumulation
20 circuit.

7. The multi-thread accumulation circuit of claim 6 further comprises an arbitration module operably coupled to the control block and the second operation unit, wherein the arbitration module receives operation codes from a plurality of thread controllers
25 corresponding to the plurality of threads, wherein the arbitration module determines order of execution of the operation codes received.

8. The multi-thread accumulation circuit of claim 7, wherein the multi-thread accumulation circuit is included in a vector engine that performs at least one of dot
30 product operations, vector multiply accumulate operations, vector addition operations, and vector multiplication operations.

5

10

15

12. A method for performing a plurality of combine and accumulate operations in a multi-thread system that supports a plurality of threads, comprising:

receiving a first set of operands corresponding to a selected thread of the plurality
5 of threads, wherein the first set of operands corresponds to a first accumulation operation for the selected thread;

combining the first set of operands to produce a first result;

10 selecting a selected accumulation register from a plurality of accumulation registers based on identity of the selected thread, wherein each accumulation register of the plurality of accumulation registers corresponds to one of the plurality of threads;

storing the first result in the selected accumulation register to produce a first
15 accumulated value;

receiving a second set of operands corresponding to the selected thread, wherein
the second set of operands corresponds to a second accumulation operation for the
selected thread;

20 combining the second set of operands to produce a second result;

combining the second result with the first accumulated value to produce a second
accumulated value; and

25 storing the second accumulated value in the selected register to produce a second accumulated result.

13. The method of claim 12, wherein combining the first set of operands includes
30 combining the first set of operands using a multiplication operation, and wherein the

combining the second set of operations further comprises combining the second set of operands using a multiplication operation.

14. The method of claim 13, wherein combining the second result with the first accumulated value further comprises combining the second result with the first accumulated value using an addition operation such that a multiply and accumulate operation for the first and second sets of operands is achieved.

15. The method of claim 12 further comprises:

receiving subsequent sets of operands corresponding to the selected thread corresponding to subsequent accumulation operations for the selected thread;

for each subsequent set of operands:

combining the subsequent set of operands to produce a subsequent result;

combining the subsequent result with a current value stored in the selected accumulation register to produce a subsequent accumulated result; and

storing the subsequent accumulated result in the selected accumulation register such that the current value stored in the selected accumulation register is updated.

16. The method of claim 12 further comprises performing combination operations corresponding to at least one additional thread of the plurality of threads subsequent to combining the first set of operands and prior to combining the second set of operands.

17. A multi-thread multiply and accumulate circuit, comprising:

an arbitration module that receives command codes corresponding to a plurality of threads, wherein at least a portion of the command codes correspond to multiply and accumulate operations, wherein the arbitration module determines an order of execution of the command codes;

a multiplier operably coupled to the arbitration module, wherein the multiplier combines a set of operands corresponding to each command code being executed to produce a product corresponding to a selected thread from which the command code being executed originated;

an adder operably coupled to the multiplier, wherein the adder combines the product of the multiplier with a second operand that is received to produce a sum corresponding to selected thread;

a plurality of accumulation registers operably coupled to the adder, wherein each of the plurality of accumulation registers corresponds to one of the plurality of threads, wherein a selected accumulation register that corresponds to the selected thread stores the sum corresponding to the selected thread; and

a selection block operably coupled to the plurality of accumulation registers and the adder, wherein the selection block selects the second operand from a set of potential operands based on control information derived from the command code being executed, wherein the set of potential operands includes values stored in each of the plurality of accumulation registers.

18. The circuit of claim 17, wherein the set of potential operands includes at least one additional operand, wherein the at least one additional operand is at least one of a constant, a state variable, and data stored in a memory structure as a result of previous operations performed by the circuit.

5

10